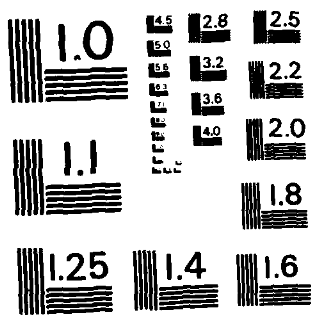
AD-A133 008 AN APPLICATION OF THE FINITE ELEMENT METHOD TO THE SOLUTION OF LOW REYNOL..(U) AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) T TRAN-CONG JUN 83 1/1

UNCLASSIFIED ARL-AERO-TM-349 F/G 9/2 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED (12)

AD-A133 008

ARL-AERO-TECH-MEMO-349 AR-002-953

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

Aerodynamics Technical Memorandum 349

AN APPLICATION OF THE FINITE ELEMENT METHOD TO THE SOLUTION OF LOW REYNOLDS NUMBER, INCOMPRESSIBLE FLOW AROUND A JOUKOWSKI AEROFOIL, WITH EMPHASIS ON AUTOMATIC GENERATION OF GRIDS

T. TRAN-CONG

Approved for Public Release

JUNE 1983

COPY No

UNCLASSIFIED 83 09 26 008

AR-002-953

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

Aerodynamics Technical Memorandum 349

AN APPLICATION OF THE FINITE ELEMENT METHOD TO THE SOLUTION OF LOW REYNOLDS NUMBER, INCOMPRESSIBLE FLOW AROUND A JOUKOWSKI AEROFOIL, WITH EMPHASIS ON AUTOMATIC GENERATION OF GRIDS

by

T. TRAN-CONG

SUMMARY

Some FORTRAN programs have been written in order to apply the Finite Element Method to the solution for low Reynolds number, incompressible flows around a Joukowski aerofoil, with emphasis on the generation of grids. These programs serve as evaluation tools and as a first step in a planned longer-term study of the Finite Element Method as applied to fluid flow problems.

POSTAL ADDRESS: Director, Aeronautical Research Laboratories,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

CONTENTS

1. INTRODUCTION

The problems of external flows around aircraft, internal flows inside propulsion units and also related problems like structural designs all require more and more accurate solutions while incorporating state of the art features involving more and more complex geometries and loadings. This is steadily becoming beyond the reach of analytical solution methods and thus necessitates the use of new computational methods. One of these is the Finite Element Method.

The Finite Element Method was initially developed and used by Zienkiewicz [1] for elasticity problems and is at the height of its development at the time of writing. The method is being studied theoretically as well as being applied to a broader and broader range of problems; its applications are found in solid mechanics, fluid mechanics, electromagnetics, etc. A reasonably short coverage of the method can be found in the book by Fenner [3].

The FORTRAN programs given at the end of this Memo have been written as an application of the Finite Element Method to a basic fluid flow problem, namely the incompressible, low Reynolds number flow about an aerofoil. A substantial part of these programs deals with the automatic generation and efficient plotting of the grids surrounding a Joukowski aerofoil.

The work reported here initiates an activity which, it is hoped, will eventually lead to a capability for aerodynamic estimation beyond that currently possible with analytical or empirical methods.

2. THEORETICAL BACKGROUND

The Finite Element Method is a method applicable to linear problems, initially developed for elasticity whereby nodal displacements in a continuum are solved for a given system of nodal forces provided these nodal forces are known for every individual mode of nodal displacement. In other words the method sets up a large system of equations, with displacements d_i's as unknowns, forces f_j's as the given constants on the right hand side of the equations, and the coefficients a_{ij}'s of the unknowns are written down from the knowledge of nodal forces resulting from each individual nodal displacement.

Our large system of equations thus takes the form:

$$\begin{array}{llll} a_{11}d_1 + a_{12}d_2 + \cdots & \cdots + a_{1n}d_n = f_1 \\ a_{21}d_1 + a_{22}d_2 + \cdots & \cdots + a_{2n}d_n = f_2 \\ \vdots & \vdots \\ a_{n1}d_1 + a_{n2}d_2 + \cdots & \cdots + a_{nn}d_n = f_n \end{array} \qquad (1)$$

where a_{ij} are the forces generated by the unit displacement d_j while other displacements are kept to zero. A high speed computer is then used to solve this system of equations when the f_j's are given. The above system is of the banded type due to the process of forming discrete elements to represent the continuum. For the individual displacement of each node the resulting non-zero nodal forces only exist at its immediate neighbour nodes, each of which must share at least one common element with the node considered. The problem of boundary conditions where displacements instead of forces are given on some boundary nodes is easily dealt with by replacing the force equations for these nodes with the simpler displacement equations which have the known values of displacements on the right hand sides and the nodal displacements on the left hand sides. A special method is then used to solve this banded system; considerable saving in computation is the direct result of the purposely designed banded character of the system.

This Finite Element Method is readily adapted to the problem of Stokes flow where the above displacements are replaced by fluid velocities at every occurrence (see [3], p. 16). The only remaining problem is the introduction of inertial forces as a kind of perturbation to the viscous forces, thus retaining the linear character of the problem.

The central problem in any Finite Element Method program is its efficiency and stability. The programs given at the end of this Memo are used in evaluating these two aspects for a certain number of different schemes available.

A substantial part of the programs given at the end of this Memo deal with the automatic generation of a grid of triangular elements around a Joukowski aerofoil. The grid is generated by wrapping an initial rectangular mesh consisting of triangular elements around a circular cylinder, this is followed by a conformal transformation to modify the cylinder into an aerofoil section. Provisions are made for varying the chord, camber, thickness of the aerofoil as well as its angle of attack. The nodal distribution can also be varied to put more nodes in the boundary layer than in the far field uniform flow.

In generating the mesh the numbering of the nodes affects the bandwidth of the system of equations to be solved. However this should not alter the computation speed of the subroutines for solving the equations. The grid with triangular elements can surround each of its nodes by the least number of six adjacent nodes, hence its use is advantageous in reducing the bandwidth of the system of equations.

The coordinate tranformation process can sometimes invert a finite sized element, i.e. change its oriented boundary into the opposite direction. This can be avoided by reducing the size of the local elements and by checking the sign of the area of every element after each coordinate transformation.

3. DESCRIPTION OF THE PROGRAMS

Program FEM sets up the grid geometry through the subroutines MESH3, MODFY3, MODFY4, ELAREA and then follows the standard finite element method described in Section 2. The matrix $[a_{ij}]$ is set up by the subroutine STIFF with the boundary conditions catered for by subroutine BCS. The system of equations is then solved by subroutine ELIMIN. The solution velocities are then used to calculate the inertial forces in INERT and modify the nodal values of forces in the next iteration for a more accurate solution of velocities. Outputs are through subroutine MSHOUT, FEMOUT, OUTPUT and SOLPLT.

Some features of these programs are:

- Since the program FEM originated from an elasticity problem it is necessary to set the Poisson ratio ν to 0.49 to simulate an incompressible flow. If this value of ν is set to 0.5 exactly the system has a number of infinite coefficients unless the problem is reformulated with one third of the equations, which are redundant, removed. Here the approximate value $\nu = 0.49$ is used to avoid the complication (see [3], p. 154).

- Subroutines STIFF and ELIMIN are for fully populated matrices. They are used here only for quick production of early results and have been replaced in subsequent work by those more suitable for banded matrices which require much less computer memory and time.

- A seam line is generated at the trailing line of the aerofoil by the MESH, MODFY3, MODFY4 subroutines to put a number of nodes there.

- In plotting the grid, two nodes of any side of an element are joined if their ordinal number increases in the anticlockwise direction around the element. This process makes the plotting computation linearly proportional to the number of elements (hence nodes) and avoids plotting any line twice.

- Although the numbering of elements and nodes does not affect the speed of the solution routines it does affect the speed of plotting the grid.

4. RESULTS

Figure 1 is the basic rectangular grid consisting of triangular elements. The numbering scheme for the 171 elements and 105 nodes is self-evident. This whole grid is then wrapped around a circle, as in Figure 2, with a scaling effect to put more nodes near to the inside. The grid then undergoes a Joukowski transformation with circulation added to become the grid in Figure 3. The Finite Element Method is then applied to the flow field using this grid. The resulting velocity field for a Reynolds number of 1.2 (based on wing chord) is plotted in Figure 4.

The results so obtained are as expected and improvements on the method are under study.

ACKNOWLEDGEMENT

The programs here were enthusiastically written and tested by Mr Martin Heinz Mann who worked with the author in his Industrial Training period with the Aeronautical Research Laboratories. Mr R.A. Feik has made a number of helpful comments on the preparation of this Memo.

REFERENCES

[1]	Zienkiewicz, O.C. and Cheung, Y.K.	The Finite Element Method in Structural and Continuum Mechanics. McGraw-Hill, 1967.
[2]	Zienkiewicz, O.C.	The Finite Element Method in Engineering Science. McGraw-Hill, 1971.
[3]	Fenner, R.T.	Finite Element Methods for Engineers. The McMillan Press Ltd., 1975.
[4]	Shen, Shan-fu	Finite Element Methods in Fluid Mechanics. Ann. Rev. Fluid Mech., 1977.

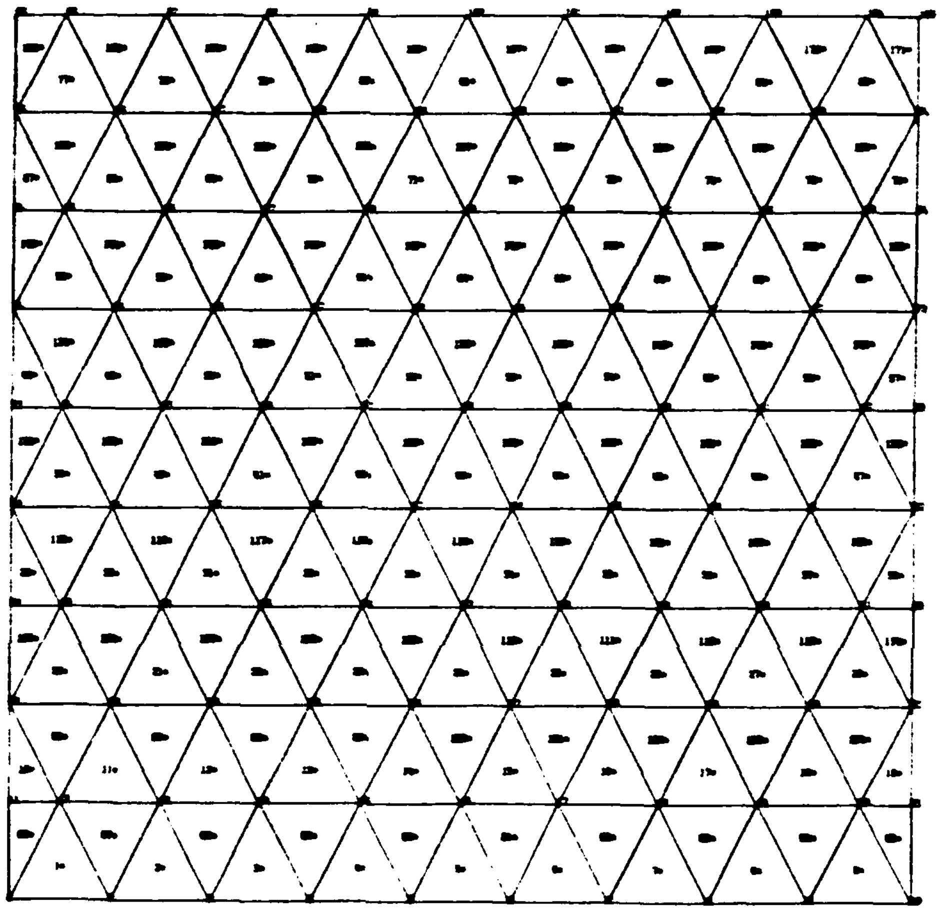


FIG.1 BASIC RECTANGULAR GRID WITH TRIANGULAR ELEMENTS

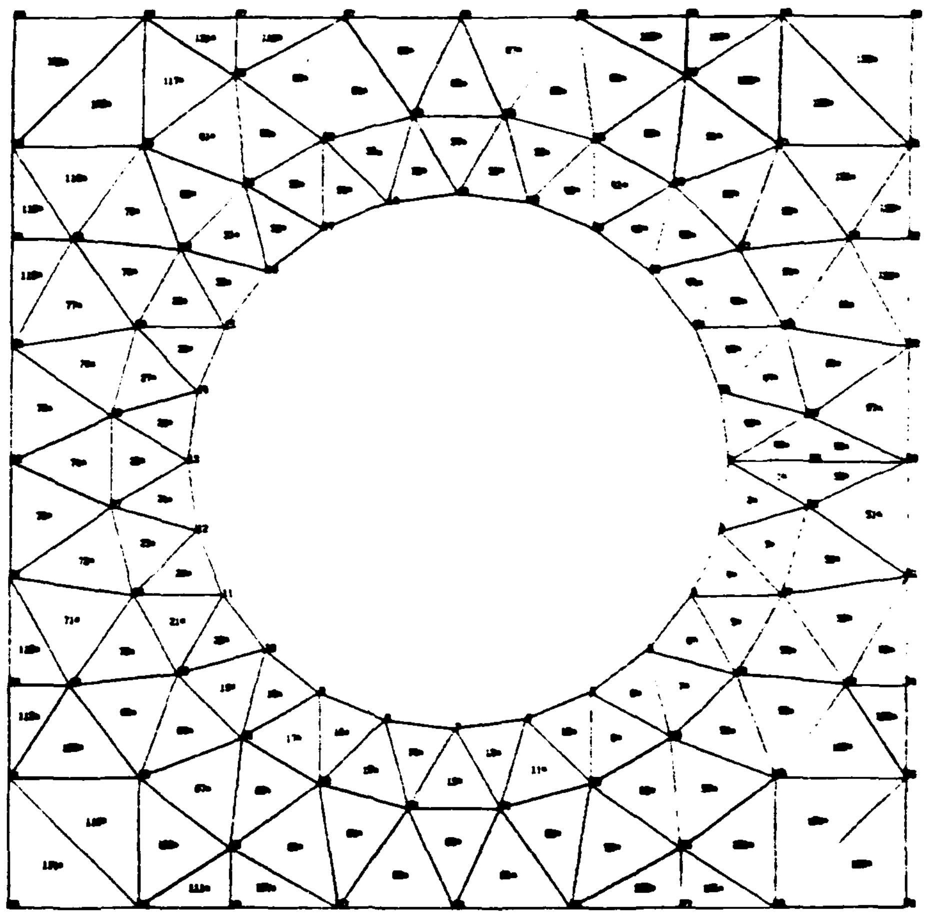


FIG.2 GRID AROUND A CYLINDER

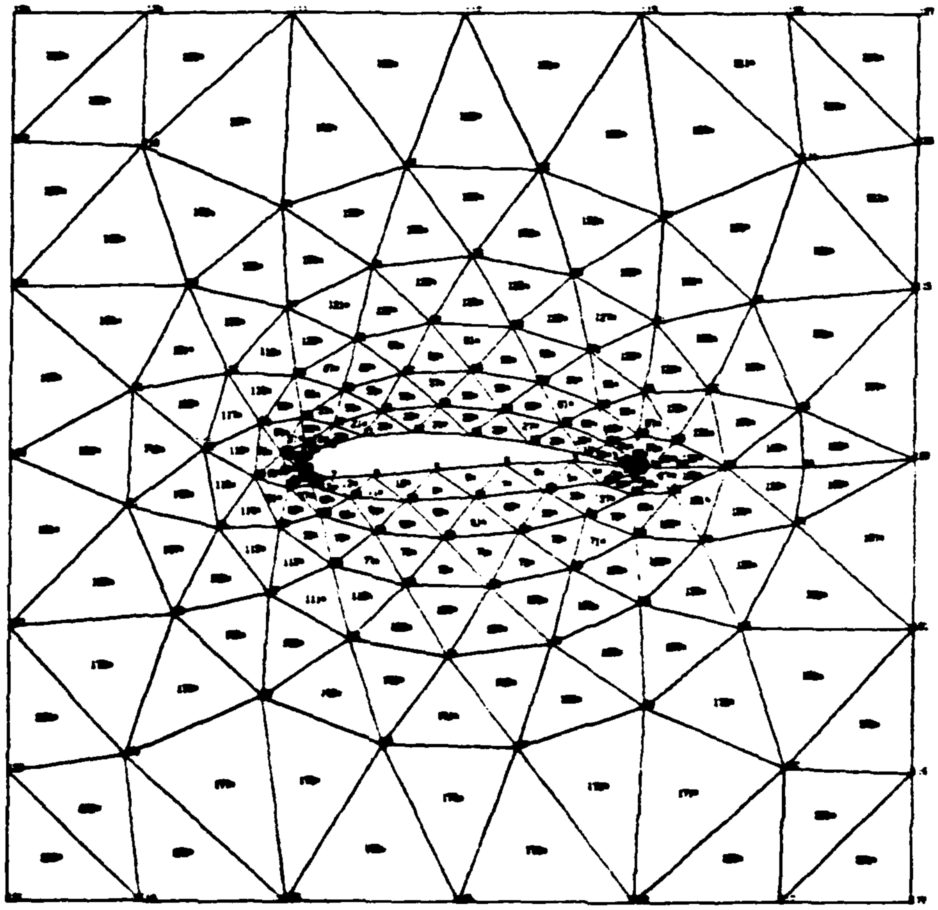


FIG.3 GRID AROUND A JOUKOWSKI AEROFOIL

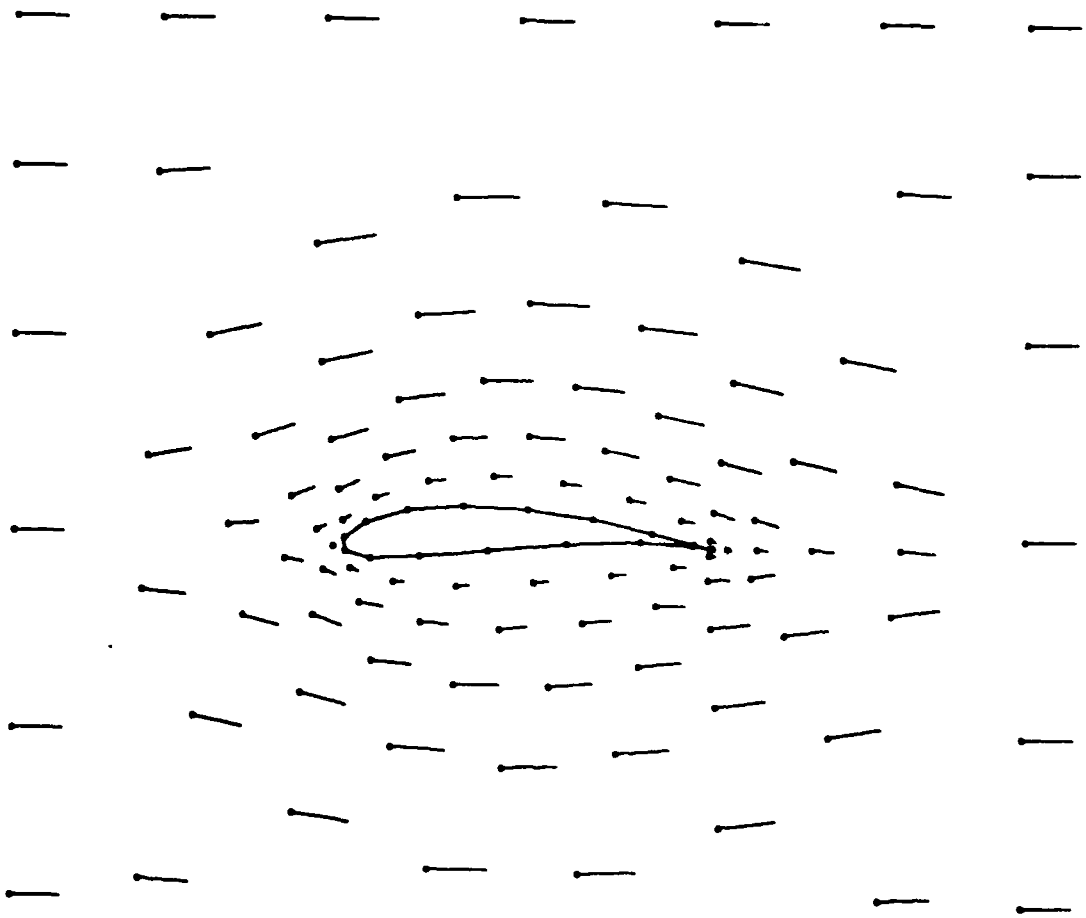


FIG.4 INCOMPRESSIBLE FLOW AT REYNOLDS NUMBER 1.2 AROUND A JOUKOWSKI AEROFOIL

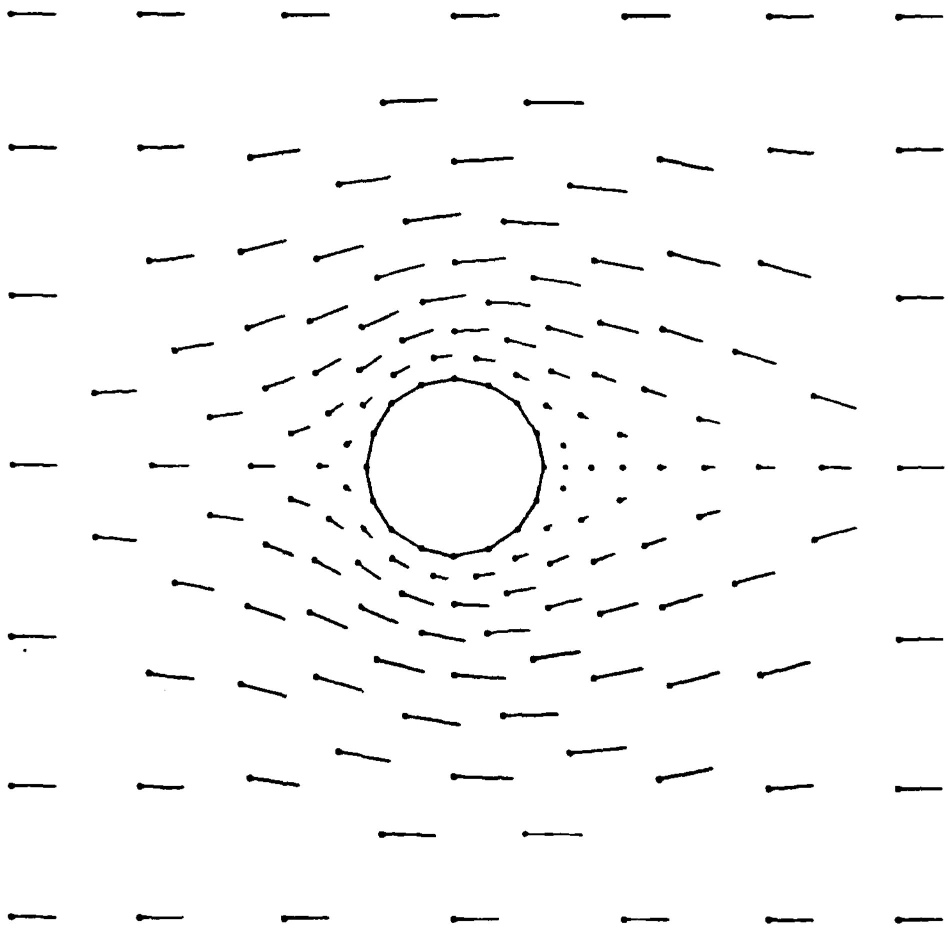


FIG.5 INCOMPRESSIBLE FLOW AROUND A CIRCULAR CYLINDER AT REYNOLDS NUMBER 6

PROGRAM LISTING

```
      PROGRAM FEM

C     PROGRAM FOR FINITE ELEMENT ANALYSIS OF TWO-DIMENSIONAL FLOW
C     ABOUT A JOUKOWSKI AEROFOIL, USING CONSTANT STRAIN TRIANGULAR
C     ELEMENTS.

      DIMENSION XSOL(300)

      REAL NU

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B
     .       /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     . FX(150),FY(150),NBP1,NBP2
     .       /CMATL/ RO

      DATA NROW,NCOL/300,300/

C     GENERATE MESH DATA.
      TYPE 10
10    FORMAT(' INPUT THE NUMBERS OF POINTS ALONG THE X AND Y AXES,',/,
     .' RESPECTIVELY, ALSO THE MESH DATA OUTPUT CONTROL PARAMETER',/,
     .' (0 IF NO MESH DATA TO BE PRINTED OUT, 1 IF ALL MESH DATA TO ',/,
     .' BE PRINTED OUT):(3I)')
      CALL MESH3
      CALL MODFY3
C     CALL MODFY4

C     COMPUTE THE ELEMENT GEOMETRIES.
      CALL ELAREA

C     OUTPUT THE MESH DATA.
      CALL MSHOUT

C     PLOT THE MESH.
      CALL MPLOT2

C     SET INITIAL VALUES OF STIFFNESSES, EXTERNAL FORCES, AND
C     VELOCITIES.
      DO 4 I=1,2*NNP
      DO 4 J=1,2*NNP
4     SSTIFF(I,J)=0.
      DO 5 I=1,NNP
      FX(I)=0.
      FY(I)=0.
      U(I)=0.
5     V(I)=0.

C     ASSEMBLE THE GLOBAL STIFFNESS MATRIX.
      CALL STIFF
      OPEN(UNIT=9,FILE='SSTIFF.TMP',MODE='BINARY')
      WRITE (9) SSTIFF
      CLOSE(UNIT=9)

C     OUTPUT THE STIFFNESS MATRIX.
      CALL STFOUT

C     INPUT DENSITY, RO.
      TYPE 66
66    FORMAT(' INPUT THE DENSITY, RO:(F)')
      ACCEPT 67,RO
67    FORMAT(F)
```

```
      OPEN(UNIT=3,FILE='FEMOUT.DAT')
      WRITE(3,23)
23    FORMAT(' SOLUTION TO FEM PROGRAM')
      CLOSE(UNIT=3)

C     SET UP ITERATIVE SOLUTION DO LOOP.
      DO 71 L=1,3

C     CALCULATE INERTIAL FORCE PER ELEMENT, AND DISTRIBUTE AMONGST THE
C     ELEMENTAL NODES.
      OPEN(UNIT=9,FILE='SSTIFF.TMP',MODE='BINARY')
      READ (9) SSTIFF
      CLOSE(UNIT=9)
      CALL INERT

C     APPLY THE BOUNDARY CONDITIONS.
      CALL BCS
      DO 90 I=1,NBP1+NBP2
      IF(I.LE.NBP1) N=NPB1(I)
      IF(I.GT.NBP1)N=NPB2(I-NBP1)
      FX(N)=U(N)
      FY(N)=V(N)
      DO 80 ICOL=1,2*NNP
      SSTIFF(2*N-1,ICOL)=0.
80    SSTIFF(2*N,ICOL)=0.
      SSTIFF(2*N-1,2*N-1)=1.
      SSTIFF(2*N,2*N)=1.
90    CONTINUE
      DO 21 I=1,NNP
      SSTIFF(2*I-1,2*NNP+1)=FX(I)
21    SSTIFF(2*I,2*NNP+1)=FY(I)

C     SOLVE THE SYSTEM OF EQUATIONS.
      NEQN=2*NNP
      TYPE 25,NEQN
25    FORMAT(' ENTERING ELIMIN',' NEQN=',I6)
      CALL ELIMIN(SSTIFF,XSOL,NEQN,NROW,NCOL,DET,AMEAN)

C     OUTPUT THE SOLUTION.
      DO 36 I=1,NNP
      U(I)=XSOL(2*I-1)
36    V(I)=XSOL(2*I)
      CALL FEMOUT
      CALL OUTPUT

71    CONTINUE

      OPEN(UNIT=9,FILE='SSTIFF.TMP')
      CLOSE(UNIT=9,DISPOSE='DELETE')

C     PLOT THE VELOCITIES.
      CALL SOLPLT
      END
      SUBROUTINE MESH3

C     SUBPROGRAM TO READ OR GENERATE A MESH OF TRIANGULAR FINITE
C     ELEMENTS.  USED IN CONJUNCTION WITH MOMFT3, THIS VERSION
C     GENERATES A CIRCULAR MESH OF MAINLY ISOSCELES TRIANGULAR
C     ELEMENTS.  NODES ARE NUMBERED IN CONTINUOUS RINGS, AND ELEMENTS
C     ARE NUMBERED IN COMPLETE ROWS, I.E. UPRIGHT AND INVERTED
C     TRIANGULAR ELEMENTS CONSIDERED ALTERNATELY.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B
```

```
C       INPUT THE NUMBERS OF POINTS ALONG THE X AND Y AXES, ALSO THE
C       MESH DATA OUTPUT CONTROL PARAMETER.
        ACCEPT 51, NXPT,NYPT,MOUT
51      FORMAT(3I)

C       COMPUTE AND TEST THE NUMBERS OF NODES AND ELEMENTS.
        MODNY=MOD(NYPT,2)
        IF(MODNY.EQ.0) NNP=NYPT*(2*NXPT-1)/2
        IF(MODNY.EQ.1) NNP=(NYPT-1)*(2*NXPT-1)/2+NXPT-1
        NEL=(NYPT-1)*(2*NXPT-1)
        IF(NNP.LE.150.AND.NEL.LE.270) GO TO 1
        TYPE 61, NNP,NEL
61      FORMAT(30HOEXCESSIVE SIZE OF MESH, NNP =,I5,8H,  NEL =,I5)
        STOP

C       DEFINE THE NODAL POINT COORDINATES.
1       I=0
        DO 3 IY=1,NYPT
        MODIY=MOD(IY,2)
        DO 2 IX=1,NXPT-1
        I=I+1
        X(I)=FLOAT(IX-1)/FLOAT(NXPT-1)
        Y(I)=FLOAT(IY-1)/FLOAT(NYPT-1)
2       IF(MODIY.EQ.0.AND.IX.GT.1) X(I)=X(I)-0.5/FLOAT(NXPT-1)
        IF(MODIY.EQ.1) GO TO 3
        I=I+1
        Y(I)=Y(I-1)
        X(I)=1.-0.5/FLOAT(NXPT-1)
3       CONTINUE

C       DEFINE THE NUMBERS OF THE THREE NODES OF EACH ELEMENT.
        M=0
        NYEL=NYPT-1
        DO 4 IY=1,NYEL
        NXEL=NXPT-1
        IF(MOD(IY,2).EQ.0) NXEL=NXPT
        DO 4 IX=1,NXEL
        M=M+1

C       MNEW REPLACES M AS THE NUMBER OF THE ELEMENT
        MNEW=2*M
        NPI(MNEW)=M
        NPJ(MNEW)=NPI(MNEW)+1
        NPK(MNEW)=NPI(MNEW)+NXPT
        IF(IX.EQ.NXEL)NPJ(MNEW)=NPJ(MNEW)-NXEL
4       IF(IX.EQ.NXPT)NPK(MNEW)=NPK(MNEW)-NXPT+1
        M1=M
        DO 5 IY=1,NYEL
        NXEL=NXPT
        IF(MOD(IY,2).EQ.0) NXEL=NXPT-1
        DO 5 IX=1,NXEL

C       MNEW REPLACES M AS THE NUMBER OF THE ELEMENT
        M=M+1
        MNEW=(M-M1)*2-1
        NPI(MNEW)=(MNEW+1)/2
        NPJ(MNEW)=NPI(MNEW)+NXPT
        NPK(MNEW)=NPI(MNEW)+NXPT-1
        IF(IX.EQ.NXPT)NPI(MNEW)=NPI(MNEW)-NXEL+1
5       IF(IX.EQ.NXEL)NPJ(MNEW)=NPJ(MNEW)-NXEL

        RETURN
        END

        SUBROUTINE MODFY3
```

```
C       SUBPROGRAM TO MODIFY A MESH TO SUIT A PARTICULAR PROBLEM.
C       THIS VERSION ADAPTS A SQUARE MESH TO A CIRCULAR ARC (RING).

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),NOUT,BLT(270)
     .          /CMPAR/ NXPT,NYPT,A,B

C       INPUT THE SCALE FACTOR AND THE MESH DIMENSIONS.
        TYPE 41
41      FORMAT(' INPUT THE SCALE FACTOR AND THE MESH DIMENSIONS:(3F)')
        ACCEPT 51,S,A,B
51      FORMAT(3F)

C       TEST FOR ACCEPTABLE BASIC MESH.
        IF(MOD(NXPT,8).EQ.1.AND.MOD(NYPT,2).EQ.1) GO TO 1
        TYPE 61
61      FORMAT(' MESH UNSUITABLE FOR PRESENT MODIFICATION')
        STOP

C       PERFORM FIRST MODIFICATION OF Y COORDINATES.
1       I=0
        HR=ALOG(S)*(NYPT-1)
        DO 2 I=1,NNP
        C=Y(I)*ALOG(S)*(NYPT-1)
        IF(S.NE.1.0)C=(EXP(C)-1.0)/(EXP(HR)-1.0)
        IF(S.EQ.1.0)C=Y(I)
        Y(I)=C
2       CONTINUE

C       PERFORM SECOND MODIFICATION TO INTRODUCE CURVATURE.
        PI=4.*ATAN(1.)
        DO 3 I=1,NNP
        R=(A+(B-A)*Y(I))/2.
        PHI=X(I)*2.0*PI
        X(I)=R*COS(PHI)+0.5*B
3       Y(I)=-R*SIN(PHI)+0.5*B


C       MODIFY COORDINATES OF POINTS NEXT TO THE END POINTS OF THE
C       OUTERMOST CIRCUMFERENTIAL ROW.
        I2=NNP-NXPT+3-(NXPT-1)/4
        I1=NNP-NXPT+1
        DO 11 K=1,4
        ARG=PI/4.-FLOAT(K)*PI/2.
        RK1=SQRT(2.)*COS(ARG)
        RK2=SQRT(2.)*SIN(ARG)
        I1=I1+(NXPT-1)/4
        I2=I2+(NXPT-1)/4
        ITEMP=I1
        I1=I2
        I2=ITEMP
        X(I1)=B/2.*(1.+RK1)
        Y(I2)=B/2.*(1.+RK2)
11      CONTINUE
        IF(NXPT.EQ.9)GOTO 22

        I1=NNP-NXPT+3-(NXPT-1)/4
        I2=NNP-NXPT+1
        DO 22 K=1,4
        ARG=PI/4.-FLOAT(K)*PI/2.
        RK1=SQRT(2.)*COS(ARG)
        RK2=SQRT(2.)*SIN(ARG)
        I1=I1+(NXPT-1)/4
        I2=I2+(NXPT-1)/4
```

```
C       DEFINE AND TEST NEW TOTAL NUMBERS OF NODES AND ELEMENTS.
        I=NNP
        NNP=NNP+(NXPT-1)/4+1-2
        M=NEL
        NEL=NEL+2*((NXPT-1)/4+1)-6
        IF(NNP.LE.150.AND.NEL.LE.270) GO TO 4
        TYPE 62,NNP,NEL
62      FORMAT(' EXCESSIVE SIZE OF MESH, NNP =',I5,'  NEL =',I5)
        STOP

C       DEFINE THE COORDINATES OF THE ADDITIONAL NODES.
4       IXMAX=(NXPT-1)/4+1-3
        DO 6 IX=1,IXMAX
        I=I+1
        II=I1+IX
        IF(IX.GT.((NXPT-1)/4+1-3)/2) GO TO 5
        X(I)=X(II)
        IF(MOD(K,2).EQ.1) X(I)=B/2.*(1.+RK1)
        Y(I)=Y(II)
        IF(MOD(K,2).EQ.0) Y(I)=B/2.*(1.+RK1)
        GO TO 6
5       X(I)=X(II-1)
        IF(MOD(K,2).EQ.0) X(I)=B/2.*(1+RK2)
        Y(I)=Y(II-1)
        IF(MOD(K,2).EQ.1) Y(I)=B/2.*(1+RK2)
6       CONTINUE
        Y(NNP)=B/2.+B/2.*RK2
        X(NNP)=B/2.+B/2.*RK1

C       DEFINE THE NODES OF THE ADDITIONAL ELEMENTS.
        M1=M
        DO 7 IX=1,IXMAX
        M=M+1
        NPI(M)=I1+M-M1-1
        NPJ(M)=NPI(M)+1
7       NPK(M)=NNP-((NXPT-1)/4+1-2)+M-M1
        M2=M
        IXMAX=IXMAX-1
        DO 8 IX=1,IXMAX
        M=M+1
        NPI(M)=I1+M-M2
        NPJ(M)=NNP-((NXPT-1)/4+1-2)+M-M2+1
8       NPK(M)=NPJ(M)-1
        NPI(NEL)=NNP-((NXPT-1)/4)/2
        NPJ(NEL)=NPI(NEL)+1
        NPK(NEL)=NNP
22      CONTINUE

C       CORRECT COORDINATES OF NODES ON MESH PERIMETER FOR SMALL
C       DISCREPANCIES.
        DO 36 I=1,NNP
        DIFFX=X(I)-B
        DIFFY=Y(I)-B
        IF(ABS(X(I)).LT.1.E-4)X(I)=0.
        IF(ABS(Y(I)).LT.1.E-4)Y(I)=0.
        IF(ABS(DIFFX).LT.1.E-4)X(I)=B
36      IF(ABS(DIFFY).LT.1.E-4)Y(I)=B

        RETURN
        END

        SUBROUTINE MODFY4

C       SUBPROGRAM TO TRANSFORM A CIRCULAR ARC TO A JOUKOWSKI AEROFOIL.
```

```
      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),NOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B

C     INPUT THE TWO ECCENTRICITIES.
      TYPE 41
41    FORMAT(' INPUT THE TWO ECCENTRITIES:(2F)')
      ACCEPT 51,X1,Y1
51    FORMAT(2F)
      X1=X1*A
      Y1=Y1*A

C     INTRODUCE TRANSLATION OF AXES.
      DO 1 I=1,NNP

C     BYPASS TRANSFORMATION OF OUTER PERIMETER NODES.
      IF((X(I).EQ.0.).OR.(X(I).EQ.B).OR.(Y(I).EQ.0.).OR.(Y(I).EQ.B))
     . GO TO 1

C     INCORPORATE CIRCULATION CORRECTION TO SATISFY KUTTA TRAILING
C     EDGE CONDITION.
      XL=X(I)
      YL=Y(I)
      PHI1=ASIN(Y1*2./A)
      R=SQRT((XL-0.5*B)*(XL-0.5*B)+
     . (YL-0.5*B)*(YL-0.5*B))
      PHI=ACOS((XL-0.5*B)/R)
      IF((YL-0.5*B).LT.0.)PHI=-PHI
      PHI=PHI+PHI1*A/(R*2.)
      XL=R*COS(PHI)+B/2.
      YL=R*SIN(PHI)+B/2.
      A1=SQRT((.5*A)**2-Y1*Y1)-X1
      XI=(XL-.5*B-X1)/A1
      ETA=(YL-.5*B-Y1)/A1

C     INTRODUCE JOUKOWSKI TRANSFORMATION.
      XISTA=XI*(1.+1./(XI*XI+ETA*ETA))
      ETASTA=ETA*(1.-1./(XI*XI+ETA*ETA))

C     REVERSE INITIAL TRANSLATION OF AXES.
      IF((X(I).NE.0.).AND.(X(I).NE.B))
     . X(I)=A1*XISTA+.5*B+X1
      IF((Y(I).NE.0.).AND.(Y(I).NE.B))
     . Y(I)=A1*ETASTA+.5*B+Y1
1     CONTINUE

      RETURN
      END

      SUBROUTINE ELAREA

C     SUBPROGRAM TO CALCULATE THE ELEMENT AREAS AND QUALITIES.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),NOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B

C     CALCULATE THE ELEMENT AREAS AND QUALITIES.
      DO 30 N=1,NEL
      I=NPI(N)
      J=NPJ(N)
      K=NPK(N)
      X1=X(J)-X(I)
      X2=X(K)-X(I)
```

```
      Y1=Y(J)-Y(I)
      Y2=Y(K)-Y(I)
      X3=X(K)-X(J)
      Y3=Y(K)-Y(J)
      SL1=SQRT(X1*X1+Y1*Y1)
      SL2=SQRT(X2*X2+Y2*Y2)
      SL3=SQRT(X3*X3+Y3*Y3)
      AREA(M)=(X1*Y2-X2*Y1)/2.
      QLTY(M)=AREA(M)/(SL1+SL2+SL3)**2.*12.*SQRT(3.)
      IF(AREA(M).GT.0.) GO TO 30
      TYPE 32,M
32    FORMAT(' ELEMENT ',I5,' HAS NEGATIVE AREA')
      STOP
30    CONTINUE

      RETURN
      END

      SUBROUTINE MSHOUT

C     SUBPROGRAM TO WRITE OUT THE GEOMETRIC DATA FOR THE MESH.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B

      IF(MOUT.EQ.0) RETURN

C     OUTPUT THE NUMBER OF ELEMENTS, NODAL POINTS AND COORDINATES.
      TYPE 61, NEL,NNP,(I,X(I),Y(I),I=1,NNP)
61    FORMAT(' GEOMETRIC DATA FOR THE MESH',//,
     .10X,'NUMBER OF ELEMENTS =',I4,//,
     .10X,'NUMBER OF NODAL POINTS =',I4,//,
     .' NODAL POINT COORDINATES',//,
     .'     I',6X,'X',8X,'Y',/
     .(1X,I5,2F9.4))

C     OUTPUT THE ELEMENT NODE NUMBERS AND AREAS.
      TYPE 62, (M,NPI(M),NPJ(M),NPK(M),AREA(M),QLTY(M),M=1,NEL)
62    FORMAT(//,' ELEMENT NODE NUMBERS,AREAS AND QUALITIES ,//,
     .'    M',4X,'I',4X,'J',4X,'K',6X,'AREA',7X,'QUALITY',/,
     .(1X,4I5,1X,E12.4,4X,F6.3))

      RETURN
      END

      SUBROUTINE MPLOT2

C     SUBPROGRAM TO PLOT THE MESH.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .       /CMPAR/ NXPT,NYPT,A,B

      DIMENSION NPNT(4),XN(4),YN(4)
      OPEN(UNIT=1,FILE='MSHPLT.OUT')

C     REPOSITION PEN.
      CALL PLOT(1,0.,5.,2)
      SCALE=10.

      DO 61 M=1,NEL
      NPNT(1)=NPI(M)
```

```
      NPNT(2)=NPJ(M)
      NPNT(3)=NPK(M)
      NPNT(4)=NPI(M)
      DO 71 K=1,3
      I1=NPNT(K)
      I2=NPNT(K+1)
      XN(1)=X(I1)
      XN(2)=X(I2)
      XN(3)=0.
      XN(4)=1./SCALE
      YN(1)=Y(I1)
      YN(2)=Y(I2)
      YN(3)=0.
      YN(4)=1./SCALE

C     CHECK WHETHER PAIR OF POINTS ARE TO BE JOINED.
      IFLAG=0
      IF(I1.LT.I2) IFLAG=1

C     CHECK WHETHER PAIR OF POINTS LIES ON MESH OUTER PERIMETER.
      IF(((XN(1).EQ.0.).AND.(XN(2).EQ.0.)).OR.
     .   ((XN(1).EQ.B).AND.(XN(2).EQ.B)).OR.
     .   ((YN(1).EQ.0.).AND.(YN(2).EQ.0.)).OR.
     .   ((YN(1).EQ.B).AND.(YN(2).EQ.B))) IFLAG=1

C     CHECK WHETHER PAIR OF POINTS LIES ON MESH INNER PERIMETER.
      IF((I1.LE.NXPT-1).AND.(I2.LE.NXPT-1)) IFLAG=1

C     JOIN APPROPRIATE POINTS.
      IF(IFLAG.EQ.1) CALL LINE(1,XN,YN,2,1,0,46)
71    CONTINUE

C     DETERMINE ELEMENT CENTROID COORDINATES.
      I=NPI(M)
      J=NPJ(M)
      K=NPK(M)
      XELC=((X(I)+X(J)+X(K))/3.)*SCALE
      YELC=((Y(I)+Y(J)+Y(K))/3.)*SCALE

C     DETERMINE NUMBER OF DIGITS IN M.
      NDM=INT(ALOG10(FLOAT(M)))+1

C     ADJUST ELEMENT NUMBER COORDINATES.
      AXELC=XELC-.5*FLOAT(NDM+1)*.06
      AYELC=YELC-.5*.07

C     NUMBER ELEMENT.
      CALL NUMBER(1,AXELC,AYELC,.07,FLOAT(M),0.,-1)
      XSTAR=AXELC+FLOAT(NDM+1)*.06-.03
C     XSTAR=XELC+FLOAT(NDM)*.06-.03
      CALL SYMBOL(1,XSTAR,YELC,.07,42,0.,-1)
61    CONTINUE

C     NUMBER NODES.
      DO 51 I=1,NNP
      TX=X(I)*SCALE
      TY=Y(I)*SCALE
      CALL NUMBER(1,TX,TY,.07,FLOAT(I),0.,-1)
51    CONTINUE
      CLOSE(UNIT=1,FILE='MSHPLT.OUT')
      RETURN
      END

      SUBROUTINE STIFF

C     SUBPROGRAM TO FORM INDIVIDUAL ELEMENT STIFFNESS MATRICES, AND
```

```
C       ASSEMBLE THE OVERALL STRUCTURE STIFFNESS MATRIX.

        REAL NU

        DIMENSION IJK(3),BMAT(3,6),D(3,3),ESTIFF(6,6)

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),NOUT,QLTY(270)
     .         /CMPAR/ NXPT,NYPT,A,B
     .         /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     .  FX(150),FY(150),NBP1,NBP2

C       INPUT THE MATERIAL PROPERTIES OF THE ELEMENTS.
        TYPE 27
27      FORMAT(' INPUT THE VALUE OF E AND NU:(2F)')
        ACCEPT 37,E,NU
37      FORMAT(2F)

        DO 50 I=1,2*NNP
        DO 50 J=1,2*NNP
50      SSTIFF(I,J)=0.
C       SET UP OVERALL ASSEMBLY LOOP.
        DO 20 N=1,NEL

C       COMPUTE THE ELEMENT GEOMETRIES.
        I=NPI(N)
        J=NPJ(N)
        K=NPK(N)
        AI(N)=-X(J)+X(K)
        AJ(N)=-X(K)+X(I)
        AK(N)=-X(I)+X(J)
        BI(N)=Y(J)-Y(K)
        BJ(N)=Y(K)-Y(I)
        BK(N)=Y(I)-Y(J)

C       STORE THE ELEMENT NODE NUMBERS IN ORDER IN ARRAY IJK.
        IJK(1)=NPI(N)
        IJK(2)=NPJ(N)
        IJK(3)=NPK(N)

C       FORM THE ELEMENT DIMENSION MATRIX, BMAT.
        DO 7 IRE=1,2
        DO 7 ICE=1,6
7       BMAT(IRE,ICE)=0.
        BMAT(1,1)=BI(N)
        BMAT(1,3)=BJ(N)
        BMAT(1,5)=BK(N)
        BMAT(2,2)=AI(N)
        BMAT(2,4)=AJ(N)
        BMAT(2,6)=AK(N)
        DO 8 ICE=1,6
        IF(MOD(ICE,2).EQ.0) BMAT(3,ICE)=BMAT(1,ICE-1)
8       IF(MOD(ICE,2).EQ.1) BMAT(3,ICE)=BMAT(2,ICE+1)

C       FORM THE ELASTIC PROPERTY MATRIX, D.
        DO 9 IRE=1,3
        DO 9 ICE=1,3
9       D(IRE,ICE)=0.
        FACT=E/((1.+NU)*(1.-2.*NU))
        D(1,1)=FACT*(1.-NU)
        D(1,2)=FACT*NU
        D(2,1)=D(1,2)
        D(2,2)=D(1,1)
        D(3,3)=FACT*.5*(1.-2.*NU)
```

```
C       FORM THE ELEMENT STIFFNESS MATRIX, ESTIFF.
        DO 10 I=1,6
        DO 10 J=1,6
        ESTIFF(I,J)=0.
        DO 10 L=1,3
        DO 10 K=1,3
10      ESTIFF(I,J)=ESTIFF(I,J)+BMAT(L,I)*D(L,K)*BMAT(K,J)*.25/AREA(M)

C       CONSTRUCT OVERALL STRUCTURE STIFFNESS MATRIX, SSTIFF.
        DO 20 IRE=1,3
        DO 20 ICE=1,3
        IROW=IJK(IRE)
        ICOL=IJK(ICE)
        SSTIFF(2*IROW-1,2*ICOL-1)=SSTIFF(2*IROW-1,2*ICOL-1)
     .  +ESTIFF(2*IRE-1,2*ICE-1)
        SSTIFF(2*IROW-1,2*ICOL)=SSTIFF(2*IROW-1,2*ICOL)
     .  +ESTIFF(2*IRE-1,2*ICE)
        SSTIFF(2*IROW,2*ICOL-1)=SSTIFF(2*IROW,2*ICOL-1)
     .  +ESTIFF(2*IRE,2*ICE-1)
        SSTIFF(2*IROW,2*ICOL)=SSTIFF(2*IROW,2*ICOL)
     .  +ESTIFF(2*IRE,2*ICE)
20      CONTINUE
        TYPE 5000
5000    FORMAT(' STIFF EXECUTED')
        RETURN
        END

        SUBROUTINE STFOUT

C       SUBPROGRAM TO OUTPUT STRUCTURE STIFFNESS MATRIX.

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),MOUT,QLTY(270)
     .        /CMPAR/ NXPT,NYPT,A,B
     .        /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     .  FX(150),FY(150),NBP1,NBP2

        TYPE 61
61      FORMAT(' INPUT THE GLOBAL STIFFNESS MATRIX OUTPUT CONTROL',/,
     .  ' PARAMETER (1 IF THE ENTIRE STIFFNESS MATRIX TO BE TYPED ',/,
     .  ' OUT, 0 IF NONE OF THE STIFFNESS MATRIX TO BE TYPED OUT):(I)')
        ACCEPT 71,KOUT
71      FORMAT(I)
        IF(KOUT.EQ.0) RETURN

        TYPE 50
50      FORMAT(/,' GLOBAL STIFFNESS MATRIX')
        DO 1 I=1,2*NNP
        TYPE 15,I
15      FORMAT(/,' ROW ',I4)
        TYPE 10,(SSTIFF(I,J),J=1,2*NNP)
10      FORMAT(6F12.7)
1       CONTINUE
        RETURN
        END

        SUBROUTINE INERT

C       SUBPROGRAM TO CALCULATE THE INERTIAL FORCE PER ELEMENT, AND
C       DISTRIBUTE IT AMONGST THE ELEMENTAL NODES.

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),MOUT,QLTY(270)
     .        /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
```

```
     .  FX(150),FY(150),NBP1,NBP2
     .        /CMATL/ RO

        DIMENSION NPNT(4)

        DO 56 I=1,NNP
        FX(I)=0.
56      FY(I)=0.

        DO 41 M=1,NEL
        NPNT(1)=NPI(M)
        NPNT(2)=NPJ(M)
        NPNT(3)=NPK(M)
        NPNT(4)=NPI(M)
        FEX=0.
        FEY=0.
        DO 81 K=1,3
        I1=NPNT(K)
        I2=NPNT(K+1)
        D1=U(I1)
        D2=U(I2)-U(I1)
        D3=V(I1)
        D4=V(I2)-V(I1)
        RL1=X(I2)-X(I1)
        RL2=Y(I2)-Y(I1)
        RL=SQRT(RL1*RL1+RL2*RL2)
        RNX=RL2/RL
        RNY=-RL1/RL
        FEX=FEX+RO*RL*(D1*((D1+D2)*RNX+(D3+.5*D4)*RNY)+D2*(D2*RNX/3.
     .  +(.5*D3+D4/3.)*RNY))
        FEY=FEY+RO*RL*(D3*((D1+.5*D2)*RNX+(D3+D4)*RNY)+D4*((.5*D1+D2/3.)
     .  *RNX+D4*RNY/3.))
81      CONTINUE
        DO 31 K=1,3
        I1=NPNT(K)
        FX(I1)=FX(I1)-FEX/3.
31      FY(I1)=FY(I1)-FEY/3.
41      CONTINUE

        RETURN
        END

        SUBROUTINE BCS

C       SUBPROGRAM TO APPLY THE BOUNDARY CONDITIONS.

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),NOUT,DLT(270)
     .        /CMPAR/ NXPT,NYPT,A,B
     .        /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     .  FX(150),FY(150),NBP1,NBP2

C       STORE NUMBERS OF NODES ON INNER MESH BOUNDARY IN ARRAY NPB1, AND
C       SET CORRESPONDING VELOCITIES TO ZERO.
        I=0
        DO 10 N=1,NXPT-1
        I=I+1
        NPB1(I)=N
        U(N)=0.
10      V(N)=0.
        NBP1=I

C       STORE NUMBERS OF NODES ON OUTER MESH BOUNDARY IN ARRAY NPB2, AND
C       SET CORRESPONDING VELOCITIES TO UNITY.
        I=0
```

```
      DO 20 N=1,NNF
      IF((X(N).NE.0.).AND.(X(N).NE.B).AND.(Y(N).NE.0.).AND.(Y(N).NE.B)
     ) GO TO 20
      I=I+1
      NPB2(I)=N
      U(N)=1.
      V(N)=0.
20    CONTINUE
      NBP2=I

      RETURN
      END

      SUBROUTINE ELIMIN(A,X,NEQN,NROW,NCOL,DET,AMEAN)

C     SUBPROGRAM FOR SOLVING SIMULTANEOUS LINEAR EQUATIONS BY GAUSSIAN
C     ELIMINATION WITH PARTIAL PIVOTING.

      DIMENSION A(NROW,NCOL),X(NROW)
      DOUBLE PRECISION SUM

      NEQN=NEQN
      IF(NEQN.LE.NROW.AND.NEQN.LE.NCOL-1) GO TO 1
      WRITE(6,61)
61    FORMAT(33H0STOP - DIMENSION ERROR IN ELIMIN)
      STOP

C     FIND MEAN COEFFICIENT MAGNITUDE.
1     AMEAN=0.
      DO 2 I=1,NEQN
      DO 2 J=1,NEQN
2     AMEAN=AMEAN+ABS(A(I,J))
      AMEAN=AMEAN/FLOAT(NEQN*NEQN)

C     COMMENCE ELIMINATION PROCESS.
      JMAX=NEQN+1
      NEQNM1=NEQN-1
      DO 6 IEQN=1,NEQNM1

C     SEARCH LEADING COLUMN OF THE COEFFICIENT MATRIX FROM THE
C     DIAGONAL DOWNWARDS FOR THE LARGEST ELEMENT AND MAKE THIS THE
C     PIVOTAL ELEMENT.
      IMIN=IEQN+1
      IMAX=IEQN
      DO 3 I=IMIN,NEQN
3     IF(ABS(A(I,IEQN)).GT.ABS(A(IMAX,IEQN))) IMAX=I
      IF(IMAX.EQ.IEQN) GO TO 5
      DO 4 J=IEQN,JMAX
      AA=A(IEQN,J)
      A(IEQN,J)=A(IMAX,J)
4     A(IMAX,J)=AA

C     ELIMINATE X(IEQN) FROM EQUATIONS (IEQN+1) TO NEQN, FIRST TESTING
C     FOR NONZERO PIVOTAL ELEMENT.
5     IF(ABS(A(IEQN,IEQN)/AMEAN).LT.1.E-8) GO TO 10
      DO 6 I=IMIN,NEQN
      FACT=A(I,IEQN)/A(IEQN,IEQN)
      DO 6 J=IMIN,JMAX
6     A(I,J)=A(I,J)-FACT*A(IEQN,J)

C     SOLVE THE UPPER-TRIANGULAR SET OF EQUATIONS BY BACK
C     SUBSTITUTION.
      IF(ABS(A(NEQN,NEQN)/AMEAN).LT.1.E-8) GO TO 10
      X(NEQN)=A(NEQN,JMAX)/A(NEQN,NEQN)
      DO 8 L=2,NEQN
      I=NEQN+1-L
```

```
      SUM=A(I,JMAX)
      IP1=I+1
      DO 7 J=IP1,NEQN
7     SUM=SUM-A(I,J)*X(J)
8     X(I)=SUM/A(I,I)

C     EVALUATE DETERMINANT OF COEFFICIENT MATRIX AND COMPARE WITH
C     ORIGINAL COEFFICIENTS.
      DETA=1.
      DO 9 I=1,NEQN
9     DETA=DETA*A(I,I)
      DET=DETA

      TYPE 72
72    FORMAT(' ELIMIN EXECUTED')

      RETURN
10    DET=0.
      TYPE 66
66    FORMAT(' ELIMIN NOT EXECUTED')
      RETURN
      END

      SUBROUTINE FEMOUT

C     SUBPROGRAM TO STORE THE SOLUTION DATA IN OUTPUT DATA FILE.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .        /CMPAR/ NXPT,NYPT
     .        /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     . FX(150),FY(150),NBP1,NBP2


      OPEN(UNIT=3,FILE='FEMOUT.DAT',ACCESS='APPEND')
      WRITE(3,11),(I,FX(I),FY(I),U(I),V(I),I=1,NNP)
11    FORMAT(//,' NODAL FORCES AND VELOCITIES',//,
     . '     N',6X,'FX',10X,'FY',11X,'U',11X,'V',/,
     . (X,I5,4E12.4))

      CLOSE(UNIT=3,FILE='FEMOUT.DAT')
      RETURN
      END

      SUBROUTINE OUTPUT

C     SUBPROGRAM TO OUTPUT THE SOLUTION DATA.

      COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     . AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     . NPJ(270),NPK(270),MOUT,QLTY(270)
     .        /CMPAR/ NXPT,NYPT
     .        /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     . FX(150),FY(150),NBP1,NBP2


      TYPE 11,(I,FX(I),FY(I),U(I),V(I),I=1,NNP)
11    FORMAT(//,' NODAL FORCES AND VELOCITIES',//,
     . '     N',6X,'FX',10X,'FY',11X,'U',11X,'V',/,
     . (X,I5,4E12.4))

      RETURN
      END

      SUBROUTINE SOLPLT
```

```
C       SUBPROGRAM TO PLOT THE NODAL VELOCITIES.

        COMMON /CMESH/ NEL,NNP,X(150),Y(150),AI(270),AJ(270),
     .  AK(270),BI(270),BJ(270),BK(270),AREA(270),NPI(270),
     .  NPJ(270),NPK(270),NOUT,DLT1(270)
     .         /CMPAR/ NXPT,NYPT,A,B
     .         /CSTIF/ SSTIFF(300,300),NPB1(50),NPB2(50),U(150),V(150),
     .  FX(150),FY(150),NBP1,NBP2

        DIMENSION XVECT(4),YVECT(4),XN(4),YN(4)
        OPEN(UNIT=2,FILE='SOLPLT.OUT')

C       REPOSITION PEN.
        CALL PLOT(2,0.,5.,2)
        SCALE=10.

        DO 1 I=1,NNP

C       COMPUTE VELOCITY VECTOR COORDINATES.
        XVECT(1)=X(I)
        XVECT(2)=X(I)+U(I)/20.
        XVECT(3)=0.
        XVECT(4)=1./SCALE
        YVECT(1)=Y(I)
        YVECT(2)=Y(I)+V(I)/20.
        YVECT(3)=0.
        YVECT(4)=1./SCALE

C       PLOT VECTORS.
        CALL LINE(2,XVECT,YVECT,2,1,0,46)
        XNODE=XVECT(1)*SCALE
        YNODE=YVECT(1)*SCALE
        CALL SYMBOL(2,XNODE,YNODE,.07,4,0.,-1)
1       CONTINUE

C       PLOT AEROFOIL.
        DO 51 I=1,NXPT-2
        XN(1)=X(I)
        XN(2)=X(I+1)
        XN(3)=0.
        XN(4)=1./SCALE
        YN(1)=Y(I)
        YN(2)=Y(I+1)
        YN(3)=0.
        YN(4)=1./SCALE
        CALL LINE(2,XN,YN,2,1,0,46)
51      CONTINUE
        XN(1)=X(NXPT-1)
        XN(2)=X(1)
        YN(1)=Y(NXPT-1)
        YN(2)=Y(1)
        CALL LINE(2,XN,YN,2,1,0,46)
        CLOSE(UNIT=2,FILE='SOLPLT.OUT')

        RETURN
        END
```

DISTRIBUTION

AUSTRALIA

Department of Defence

Central Office

Chief Defence Scientist)
Deputy Chief Defence Scientist) (1 copy)
Superintendent, Science and Technology Programmes)
Controller, Projects and Analytical Studies)
Defence Science Representative (UK) (Doc Data sheet only)
Counsellor, Defence Science (USA) (Doc Data sheet only)
Defence Central Library
Document Exchange Centre, D.I.S.B. (17 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratories

Director
Library
Superintendent - Aerodynamics Division
Divisional File - Aerodynamics
Author: T. Tran-Cong

Materials Research Laboratories

Director/Library

Defence Research Centre

Library

Navy Office

Navy Scientific Adviser

Army Office

Army Scientific Adviser

Air Force Office

Air Force Scientific Adviser

Central Studies Establishment

Information Centre

.../contd.

DISTRIBUTION (CONTD.)

Department of Defence Support

Government Aircraft Factories

Manager
Library

Universities and Colleges

Sydney	Dr G.P. Steven, Dept. of Aeronautical Engineering

SPARES (10 copies)

TOTAL (50 copies)

Department of Defence

DOCUMENT CONTROL DATA

1. a. AR No	1. b. Establishment No	2. Document Date	3. Task No
AR-002-953	ARL-AERO-TECH-MEMO-349	June 1983	DST 82/031

4. Title	5. Security a. document	6. No Pages
AN APPLICATION OF THE FINITE ELEMENT METHOD TO THE SOLUTION OF LOW REYNOLDS NUMBER, INCOMPRESSIBLE FLOW AROUND A JOUKOWSKI AEROFOIL, WITH EMPHASIS ON AUTOMATIC GENERATION OF GRIDS.	UNCLASSIFIED b. title: U c. abstract: U	4 7. No Refs: 4

8. Author(s)	9. Downgrading Instructions
T. TRAN-CONG	-

10. Corporate Author and Address	11. Authority (as appropriate) a.Sponsor b.Security c.Downgrading d.Approval
Aeronautical Research Laboratories P.O. Box 4331, Melbourne, Victoria. 3001.	-

12. Secondary Distribution (of this document)

Approved for public release.

Overseas enquirers outside stated limitations should be referred through ASDIS, Defence Information Services Branch, Department of Defence, Campbell Park, CANBERRA ACT 2601

13. a. This document may be ANNOUNCED in catalogues and awareness services available to ...

No limitations.

13. b Citation for other purposes (ie casual announcement) may be ~~select~~, unrestricted ~~(or) as for 13 a.~~

14. Descriptors		15. COSATI Group
Viscous flow.	Joukowski airfoil.	0101
Incompressible flow.		1201
Computation.		0902
Finite element analysis.		

16. Abstract

Some FORTRAN programs have been written in order to apply the Finite Element Method to the solution for Low Reynolds number, incompressible flows around a Joukowski aerofoil, with emphasis on the generation of grids. These programs serve as evaluation tools and as a first step in a planned longer-term study of the Finite Element Method as applied to fluid flow problems.

PF 65

This page is to be used to record information which is required by the Establishment for its own use but which will not be added to the DISTIS data base unless specifically requested.

16. Abstract (Contd)		
17. Imprint Aeronautical Research Laboratories, Melbourne, Australia.		
18. Document Series and Number AERODYNAMICS TECHNICAL MEMORANDUM 349	19. Cost Code 527770	20. Type of Report and Period Covered -
21. Computer Programs Used FEM (FORTRAN)		
22. Establishment File Ref(s) -		

END

DATE
FILMED

10 83

DT